PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

Application Number	10/648,600
Filing Date	August 25, 2003
First Named Inventor	Namit Jain
Art Unit	2165
Examiner Name	Radtke, Mark A.
Attorney Docket Number	50277-2235

ENCLOSURES (Check all that apply)

- ☒ Fee Transmittal Form
- ☒ Fee Attached
- ☐ Amendment/Reply
- ☐ After Final
- ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
- ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation
- ☐ Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☒ Other Enclosure(s) (please Identify below): Appeal Brief and Return receipt postcard

Remarks

The Director is hereby authorized to charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 to Deposit Account Number **50-1302**
Deposit Account Name: Hickman Palermo Truong & Becker LLP

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Hickman Palermo Truong & Becker LLP		
Signature			
Printed name	Christian A. Nicholes		
Date	4/25/2007	Reg. No.	50,266

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Annette Valdivia		
Typed or printed name	Annette Valdivia	Date	4/25/07

SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

PTO/SB/17 (12-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($)500.00

Complete if Known	
Application Number	10/648,600
Filing Date	August 25, 2003
First Named Inventor	Namit Jain
Examiner Name	Radtke, Mark A.
Art Unit	2165
Attorney Docket Number	50277-2235

METHOD OF PAYMENT (check all that apply)

☒ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

Deposit Account Deposit Account Number: 50-1302 Deposit Account Name: Hickman Palermo Truong & Becker LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☐ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, **except for the filing fee**

☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☒ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
Application Type	Fee ($)	Small Entity Fee ($)	Fee ($)	Small Entity Fee ($)	Fee ($)	Small Entity Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 (including Reissues	50	25
Each independent claim over 3 (including Reissues)	200	100
Multiple dependent claims	360	180

Total Claims _____ - 20 or HP = _____ x _____ = _____ Extra Claims Fee ($) Fee Paid ($)

HP = highest number of total claims paid for, if greater than 20.

Indep. Claims _____ - 3 or HP = _____ x _____ = _____ Extra Claims Fee ($) Fee Paid ($)

HP = highest number of independent claims paid for, if greater than 3

Multiple Dependent Claims: Fee ($) _____ Fee Paid ($) _____

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets _____ - 100 = _____ / 50 = _____ (round up to a whole number) x _____ = _____ Extra Sheets Number of each additional 50 or fraction thereof Fee ($) Fees Paid ($)

4. OTHER FEE(S)

Non-English Specification, $130 fee (no small entity discount)

Other (e.g., late filing surcharge) 1402-appeal brief

Fees Paid ($) 500.00

SUBMITTED BY

Signature	[signature]	Registration No. 50,266 (Attorney/Agent)	Telephone 408.414.1224
Name (Print/Type)	Christian A. Nicholes		Date 4/25/2007

SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450. *If you need assistance in completing the form, call 1-800-PTO-9199 and select opti*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of:

Namit JAIN

Serial No.: 10/648,600

Filed on: August 25, 2003

For: DIRECT LOADING OF SEMISTRUCTURED DATA

Confirmation No.: 3748

Examiner: RADTKE, Mark A.

Group Art Unit No.: 2165

MS Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed on February 9, 2007. A Pre-Appeal Brief Request for Review was filed with the Notice of Appeal. The Notice of Panel Decision from Pre-Appeal Brief Review was mailed on March 26, 2007. Therefore, the period in which this Appeal Brief may be filed extends to April 26, 2007.

I. REAL PARTY IN INTEREST

Oracle International Corporation is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

The present application is related to U.S. Patent Application Serial No. 10/648,749, which is also under appeal before the Board of Patent Appeals and Interferences. An appeal

brief was filed in the latter matter on February 12, 2007. The Honorable Board has not yet handed down a decision in that appeal.

III. STATUS OF CLAIMS

Claims 1-26 have been finally rejected and are the only subjects of this appeal.

IV. STATUS OF AMENDMENTS

Claims 1-13 were not amended after the Final Office Action. Claims 14-26 were amended after the Final Office Action. The amendments to Claims 14-26 were entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application contains independent Claims 1 and 12, which are summarized below. The claims summarized below are annotated to cross-reference features of the claims to specific examples of those features disclosed in the specification. However, the annotations are not intended to limit the scope of the recited features to those specific examples to which the annotations refer.

Claim 1 recites (with added reference annotations in parenthesis) a method (FIG. 6A through FIG. 6D) of storing data (FIG. 5, semistructured data 518) into a database (FIG. 5, database 506), the method comprising:

a client application (FIG. 5, client application 502) receiving (FIG. 6B, step 610; paragraphs [0038], [0068], and [0099]) data (FIG. 5, semistructured data 518; paragraphs [0023]-[0024]);

determining (FIG. 6B, step 612; paragraphs [0038], [0069], and [0100]) one or more routines (FIG. 5, routines 510AA-AN) that are associated (paragraphs [0037], [0096], and [0098]) with a type (paragraph [0015]) of said data, wherein said

one or more routines are implemented (paragraphs [0095] and [0097]) by a program (FIG. 5, type implementor 508A) that is external (paragraphs [0021], [0037], and [0045]) to both said client application and a database server (FIG. 5, database server 504) that manages said database;

invoking (FIG. 6B, step 614; paragraphs [0039], [0071], and [0101]) said one or more routines;

in response to said one or more routines being invoked (paragraphs [0039] and [0102]), said program performing steps comprising:

determining (FIG. 6D, step 626; paragraphs [0039], [0072]-[0079], [0090], [0102], and [0105]) one or more first values that are specified in said data, wherein said one or more first values correspond (paragraphs [0015] and [0043]) to one or more attributes (paragraph [0015]) of said type; and

determining (FIG. 6D, step 628; paragraphs [0040]-[0042], [0102], and [0106]) one or more second values that correspond to one or more hidden columns (FIG. 1, hidden columns 108A-N; paragraphs [0047]-[0052] and [0074]) of one or more tables in said database;

generating (FIG. 6D, steps 626, 628, and 634; paragraphs [0044], [0080], and [0110]-[0111]), based on (paragraph [0051]) said one or more first values and said one or more second values (e.g., in arrays 516A-N of FIG. 5), a data stream that conforms (paragraphs [0017] and [0045]) to a format of data blocks of said database; and

writing (FIG. 6D, step 638; paragraphs [0044], [0080], and [0112]-[0113]) said data into one or more data blocks in said database.

Claim 12 recites (with added reference annotations in parenthesis) a method (FIG. 6A through FIG. 6D) of storing data (FIG. 5, semistructured data 518; paragraphs [0023]-[0024]) into a database (FIG. 5, database 506), the method comprising:

a client application (FIG. 5, client application 502) receiving (FIG. 6B, step 610; paragraphs [0038], [0068], and [0099]) data (FIG. 5, semistructured data 518; paragraphs [0023]-[0024]) that conforms to a first type definition (paragraph [0015]) that indicates two or more first attributes, wherein at least one of said two or more first attributes is of a type that is defined by a second type definition (paragraphs [0043], [0054], [0064], and [0077]) that indicates two or more second attributes (paragraph [0083]);

determining (FIG. 6B, step 612; paragraphs [0038], [0069], and [0100]) one or more first routines (FIG. 5, routines 510AA-AN) that are associated (paragraphs [0037] and [0096]) with said first type definition, wherein said one or more first routines are external (paragraphs [0021], [0037], and [0045]) to both said client application and a database server (FIG. 5, database server 504) that manages said database;

calling (FIG. 6B, step 614; paragraphs [0039], [0071], and [0101]) said one or more first routines;

in response to one or more calls (paragraphs [0039] and [0102]) to said one or more first routines:

creating (FIG. 6B, step 616; paragraph [0102]) a first data structure (FIG. 5, array 516A; FIG. 1, array 104; paragraph [0072]) with two or more first elements that correspond to said two or more first attributes; and

populating (FIG. 6D, step 626; paragraph [0105]) said two or more first elements with two or more first values that are specified in said data (paragraphs [0074]-[0076]), wherein said two or more first values correspond to said two or more first attributes;

calling (FIG. 6C, step 622; paragraph [0103]) one or more second routines (FIG. 5, routines 510BA-BN; paragraph [0077]) that are associated (paragraphs [0037] and [0096]) with said second type definition;

in response to one or more calls (paragraph [0077]) to said one or more second routines:

creating (FIG. 6C, step 624; paragraph [0104]) a second data structure (FIG. 5, array 516B; paragraphs [0077]-[0078]) with two or more second elements that correspond to said two or more second attributes; and

populating (FIG. 6D, step 630; paragraph [0107]) said two or more second elements with two or more second values that are specified in said data (paragraphs [0077]-[0078]), wherein said two or more second values correspond to said two or more second attributes;

generating (FIG. 6D, step 634; paragraphs [0044], [0080], and [0110]-[0111]), based on (paragraph [0051]) said first data structure and said second data structure, a data stream that conforms (paragraphs [0017] and [0045]) to a format of data blocks of said database; and

writing (FIG. 6D, step 638; paragraphs [0044], [0080], and [0112]-[0113]) said data into one or more data blocks in said database.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-4, 6, 7, 10, 11, 14-17, 19, 20, 23, and 24 stand rejected under 35 U.S.C. § 102(b) as being anticipated, allegedly, by U.S. Patent No. 6,085,198 ("Skinner").

2. Claims 12 and 25 stand rejected under 35 U.S.C. § 102(b) as being anticipated, allegedly, by Skinner.

3. Claims 5 and 18 stand rejected under 35 U.S.C. § 102(b) as being anticipated, allegedly, by Skinner.

4. Claims 8 and 21 stand rejected under 35 U.S.C. § 102(b) as being anticipated, allegedly, by Skinner.

5. Claims 9 and 22 stand rejected under 35 U.S.C. § 102(b) as being anticipated, allegedly, by Skinner.

6. Claims 13 and 26 stand rejected under 35 U.S.C. § 102(b) as being anticipated, allegedly, by Skinner.

VIII. ARGUMENTS

A. The Features of Claims 1-4, 6, 7, 10, 11, 14-17, 19, 20, 23, and 24 Are Not Disclosed, Taught, or Suggested by Skinner

Among other features, Claim 1 recites, "determining one or more second values **that correspond to one or more hidden columns of one or more tables in said database**." However, Skinner does not disclose, teach, or suggest anything about "hidden columns" of a database table. The portion of Skinner that refers to database table creation begins at col. 37,

line 9, and ends at col. 39, line 6. The notion of "hidden columns" is not found anywhere in this text.

The Examiner relies on col. 20, lines 24-27 of Skinner as allegedly disclosing "hidden columns of a database table." However, this portion of Skinner actually refers to "private" and "protected" states of **class elements**. As will be seen from the discussion below, there is absolutely no reason to assume that a value of a "private class element" must correspond to a "hidden column of a database table."

Skinner, col. 20, lines 24-27, reads, "MetaMember 502 comprises an integer data structure containing 'myPrivateFlag.' 'myPrivateFlag' describes the private and protected state of the class element described by MetaMember 502."

Despite the private accessibility mode of the class element described by MetaMember 502 within an object-oriented class, and assuming that this element even corresponds to a database table column (which Skinner does **not** indicate), it does not logically follow from the accessibility mode of this element that such a database column must be a "hidden" column. Although this element might need to be declared as "private" in order to prevent subclasses of the class in which the element is declared from accessing the element, there is no reason to assume that the data stored in the element would also need to be obscured from query results or made invisible to a user by storing the data in a hidden column of a database table. As used in Skinner, "private" refers to the fact that the element cannot be accessed by subclasses that extend the class in which the element is declared; in this context, "private" does **not** mean that the element or its associated data should be obscured from a user. An element's accessibility to subclasses really has nothing at all to do with whether data that corresponds to that element should be stored in a hidden column of a database table.

The fact that an element is declared to be "private" or "protected" within a class does not mean that a database table column that stores data associated with that element is "hidden" by definition. A database column that is *not* hidden might store data that is associated with an element that is declared to be "private." Conversely, a hidden column might store data that is associated with an element that *not* declared to be "private."

Thus, the rejection of Claim 1 appears to be based on a misunderstanding of what the definition of a "private" or "protected" class element is. A definition of these terms in the object-oriented programming language context is given, for example, in *Essentials of the Java Programming Language: A Hands-On Guide, Part 2*, which can be viewed on-line at "java.sun.com/developer/onlineTraining/Programming/BasicJava2/oo.html#access". The fact that *Essentials* is made available by the same entity to which Skinner is assigned (Sun Microsystems, Inc.) should give *Essentials* some measure of credibility.

In the subsection titled "Fields and Methods," *Essentials* says:

> Fields and methods can be declared **`private`**, **`protected`**, `public`, or `package`. If no access level is specified, the field or method access level is `package` by default.
> . . . **`private`: A private field or method is accessible only to the class in which it is defined**. . . . **`protected`: A protected field or method is accessible to the class itself, its subclasses, and classes in the same package**. . . . `public`: A public field or method is accessible to any class of any parentage in any package. . . . `package`: A package field or method is accessible to other classes in the same package.

Therefore, although the "private" or "protected" declaration of a class element influences which **other classes** can access (i.e., inherit) that class element, the "private" or "protected" declaration has absolutely no bearing on whether the values of that element are **visible** to any **user**. Since the "private" declaration of a class element has absolutely no bearing on whether the values of that class element are visible to **users**, there is no reason for a "private" class

element to correspond to a hidden column of a database table or for the values of such a class element to be stored in such a hidden column.

Indeed, when an object is instantiated from a class that declares one of its elements to be "private," the value of the private element of that object **is as visible** to users as the values of elements that have been declared to be "public." A class element's values are **not** hidden from **users** even if that class element has been declared to be "private."

With an accepted understanding of the meaning of the terms "private" and "protected" as used in Skinner, it should be clear why the "private" or "protected" nature of a class element has nothing to do with hidden columns of database tables.

The Examiner takes the position that values of hidden columns and values of private class members are both "hidden" from a user's point of view, and that, as a result, private class members are essentially the same as hidden columns. However, the discussion above shows that the values of private class members are **not**, by definition, hidden from a user's point of view. Furthermore, although it is true that values of hidden columns of a database table are hidden from a user when the user queries a table that contains such columns, it does not logically follow that everything that is hidden from a user must necessarily reside in a hidden column of a database table.

Private members of a class simply are **not** the same as hidden columns of a database table, and anyone of ordinary skill in the arts of object-oriented programming and databases knows that they are not the same. Skinner cannot anticipate Claim 1 under 35 U.S.C. § 102 unless Skinner **actually discloses** all of the features of Claim 1. Skinner simply does not do so. The Examiner's burden of showing that Skinner actually discloses exactly the same features as are recited in Claim 1 simply has not been met. It is not enough for Skinner to

disclose something that the Examiner thinks might have some characteristics in common with hidden columns of a database table.

Although there are, no doubt, innumerable things that have one or more characteristics in common with hidden columns of a database table, these common characteristics do not make these things the same as, or substitutable for, hidden columns of a database table. From the perspective of one of ordinary skill in the art, it is a **clear** error of fact to say that private members of an object-oriented class are the same thing as hidden columns of a database table, regardless of whether both involve some "hiding" properties.

Therefore, Skinner does not disclose, teach, or suggest "determining one or more second values **that correspond to one or more hidden columns of one or more tables in said database**" as recited in Claim 1. Therefore, Claim 1 is patentable over Skinner under 35 U.S.C. § 102(b).

By virtue of their dependence from Claim 1, Claims 2-4, 6, 7, 10, 11, 14-17, 19, 20, 23, and 24 include the features of Claim 1 that are distinguished from Skinner above. As a result, Claims 2-4, 6, 7, 10, 11, 14-17, 19, 20, 23, and 24 are patentable over Skinner under 35 U.S.C. § 102(b) for at least the reasons discussed above in connection with Claim 1. The rejection of Claims 1-4, 6, 7, 10, 11, 14-17, 19, 20, 23, and 24 should be reversed.

B. The Features of Claims 12 and 25 Are Not Disclosed, Taught, or Suggested by Skinner

Among other features, Claim 12 recites, "a client application receiving data that conforms to a first type definition that indicates two or more first attributes, wherein at least one of said two or more first attributes is of a type that is defined by a second type definition that indicates two or more second attributes." In other words, Claim 12 requires that at least

one of the attributes of the type to which the data conforms must itself be of a type that comprises multiple attributes. The Examiner alleges that Skinner discloses this feature in col. 16, lines 48-49, which read, "In step 400, the schema describing the data classes to be used in the system is obtained." However, the cited text does not indicate that the data classes have the specific qualities of the data recited in Claim 12.

The Examiner responds by saying that Skinner discloses multiple inheritance. However, Claim 12 does not say "inheritance." Instead, Claim 12 refers to two separate type definitions. Significantly, **one of the attributes of** the first type definition must be of a type that is defined by the **second type definition**. Also significantly, the second type definition **must** indicate **two or more attributes**. So, Claim 12 cannot be anticipated by Skinner under 35 U.S.C. § 102(b) if Skinner does not disclose, at least, a type that has two or more attributes, at least one of which attributes must be of a type that also has two or more attributes. Regardless of whether Skinner discloses multiple inheritance, the disclosure of multiple inheritance does not imply the disclosure of type definitions having the qualities recited in Claim 12. One may define a class that inherits from multiple classes **without** ever defining type definitions of the kind recited in Claim 12.

Therefore, Skinner does not disclose, teach, or suggest "a client application receiving data that conforms to a first type definition that indicates two or more first attributes, wherein at least one of said two or more first attributes is of a type that is defined by a second type definition that indicates two or more second attributes" as recited in Claim 12. Therefore, Claim 12 is patentable over Skinner under 35 U.S.C. § 102(b).

By virtue of its dependence from Claim 12, Claim 25 includes the features of Claim 12 that are distinguished from Skinner above. As a result, Claim 25 is patentable over

Skinner under 35 U.S.C. § 102(b) for at least the reasons discussed above in connection with Claim 12. The rejection of Claims 12 and 25 should be reversed.

C. The Features of Claims 5 and 18 Are Not Disclosed, Taught, or Suggested by Skinner

Claim 5 depends from Claim 1, and further recites "wherein at least one of said one or more second values **describe a position** of said one or more first values relative to other values in said data." For example, such a positional descriptor is discussed in paragraph [0042] of the application. The Examiner alleges that Skinner discloses this feature in col. 20, lines 7-9, which read "'myPassedMethods' is a Vector of MetaMethod instances passed by other MetaClasses to the current MetaClass instance." However, this text says nothing about one value describing a **position** of another value within data.

Therefore, Skinner does not disclose, teach, or suggest "wherein at least one of said one or more second values **describe a position** of said one or more first values relative to other values in said data" as recited in Claim 5. Therefore, Claim 5 is patentable over Skinner under 35 U.S.C. § 102(b).

By virtue of its dependence from Claim 5, Claim 18 includes the features of Claim 5 that are distinguished from Skinner above. As a result, Claim 18 is patentable over Skinner under 35 U.S.C. § 102(b) for at least the reasons discussed above in connection with Claim 5. The rejection of Claims 5 and 18 should be reversed.

D. The Features of Claims 8 and 21 Are Not Disclosed, Taught, or Suggested by Skinner

Claim 8 depends from Claim 1 and further recites "wherein said generating and said writing are performed without causing a Structured Query Language (SQL) engine to load

said data." Such an approach is discussed in paragraph [0045] of the present application, for example. The "writing" to which Claim 8 refers is "writing said data into one or more data blocks **in said database**" as recited in Claim 1.

The Examiner alleges that Skinner discloses this feature in col. 18, lines 8-12, which read "By compiling the schema file and executing the method calls, the schema metadata may be extracted and loaded directly into the desired data structures, with the desired flags automatically set in accordance with the specified methods." However, this text does **not** refer to loading data into a **database**. No part of this text refers to a database. In this text, "desired data structures" does not refer to structures within a database. Instead, this text refers to storing metadata (extracted from a schema) into "metadata structures;" the metadata thus extracted and stored may be applied later to "table generation processes" (col. 17, lines 56-61). The "loading" is performed **prior** to the generation of tables in a database; therefore, the loading cannot refer to the loading of data **into** a database.

Thus, even if the schema metadata is "extracted and loaded directly" into the "desired data structures," and if the metadata were later stored in a database, there is no reason to believe that the storage into the database would be accomplished through any mechanisms other than SQL commands issued to an SQL engine. Skinner does not indicate, anywhere, that data is written into a database "without causing a Structure Query Language (SQL) engine to load said data."

Therefore, Skinner does not disclose, teach, or suggest "wherein said generating and said writing are performed without causing a Structured Query Language (SQL) engine to load said data" as recited in Claim 8. Therefore, Claim 8 is patentable over Skinner under 35 U.S.C. § 102(b).

By virtue of its dependence from Claim 8, Claim 21 includes the features of Claim 8 that are distinguished from Skinner above. As a result, Claim 21 is patentable over Skinner under 35 U.S.C. § 102(b) for at least the reasons discussed above in connection with Claim 8. The rejection of Claims 8 and 21 should be reversed.

E. The Features of Claims 9 and 22 Are Not Disclosed, Taught, or Suggested by Skinner

Claim 9 depends from Claim 1 and further recites "wherein determining said one or more routines comprises locating addresses of one or more routines that are in a same entry as an identity of said type." A dispatch table that associates (a) type identities with (b) memory addresses of routines is discussed in paragraph [0069] of the present application, for example.

The Examiner alleges that Skinner discloses this feature in col. 16, lines 40, which merely indicates that attribute names may have associated data types. The Appellants do not dispute that attributes may have data types. Indeed, the Appellants admitted as much in paragraph [0015] in the "background" section of the present application ("Each attribute is also of a type."). However, the cited portion of Skinner does **not** say anything about an "entry" that contains both (a) an "identity of a type" and (b) "**addresses** of **routines**."

Even if Skinner disclosed a table in which each entry associated an attribute with that attribute's data type, such a table would still have nothing to do with **addresses** of **routines** that can be invoked. An "address of a routine" is not the same as a "data type." Skinner does not disclose, teach, or suggest "wherein determining said one or more routines comprises locating addresses of one or more routines that are in a same entry as an identity of said type"

as recited in Claim 9. Therefore, Claim 9 is patentable over Skinner under 35 U.S.C. § 102(b).

By virtue of its dependence from Claim 9, Claim 22 includes the features of Claim 9 that are distinguished from Skinner above. As a result, Claim 22 is patentable over Skinner under 35 U.S.C. § 102(b) for at least the reasons discussed above in connection with Claim 9. The rejection of Claims 9 and 22 should be reversed.

F. The Features of Claims 13 and 26 Are Not Disclosed, Taught, or Suggested by Skinner

Claim 13 depends from Claim 12 and further recites "generating **a set identifier** that is associated with one of said one or more first elements; and **populating a plurality of elements in said second data structure with said set identifier**." An approach that populates a **plurality** of rows in a child element database table with a set identifier that is also stored in a row in a parent element database table is discussed in paragraph [0063] of the present application, for example.

The Examiner alleges that Skinner discloses the generation of the set identifier in col. 20, lines 29-31. Thus, the Examiner apparently alleges that the "set identifier" is analogous to the value of Skinner's "myName" string structure, which is discussed in those lines. If Skinner truly did disclose the features of Claim 13, then Skinner would need to disclose that the value of Skinner's "myName" string structure was populated in a **plurality of elements** in some other "second" data structure.

It is unclear what aspect of Skinner the Examiner means to be analogous to the "second data structure" of Claim 13. Instead of providing detailed reasoning and citing a specific structure of Skinner being analogous to the "second data structure" of Claim 13, the

Examiner generally refers to Skinner's FIGs. 5A and 5B, which illustrate multiple different metadata structures. One can only speculate as to which of these metadata structures the Examiner means to refer.

Classes "MetaAttribute" and "MetaMethod" apparently inherit the "myName" structure from class "MetaMember." However, even if instances of each of these classes were created, and even if the value of "myName" were to be populated in each instance, there still would not be **any one** instance in which **a plurality of elements** were populated with that value; "myName" only occurs **once** in each instance.

Class "MetaParameter" also contains a "myName" structure. However, even if an instance of "MetaMember" and an instance of "MetaParameter" were created, and even if the same value were inserted into the "myName" structures of each instance, there still would not be **any one** instance in which **a plurality of elements** were populated with that value; "myName" only occurs **once** in each instance.

Skinner does not appear to disclose any single data structure in which **a plurality** of elements could be populated with a particular set identifier, as required by Claim 13. Therefore, Skinner does not disclose, teach, or suggest "generating **a set identifier** that is associated with one of said one or more first elements; and **populating a plurality of elements in said second data structure with said set identifier**" as recited in Claim 13. Therefore, Claim 13 is patentable over Skinner under 35 U.S.C. § 102(b).

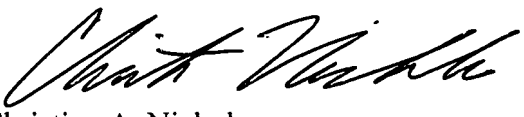
By virtue of its dependence from Claim 13, Claim 26 includes the features of Claim 13 that are distinguished from Skinner above. As a result, Claim 26 is patentable over Skinner under 35 U.S.C. § 102(b) for at least the reasons discussed above in connection with Claim 13. The rejection of Claims 13 and 26 should be reversed.

IX. CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that the rejections of Claims 1-26 lack the requisite factual and legal bases. Appellants respectfully request that the Honorable Board **reverse** the rejections of Claims 1-26.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP



Date: April 25, 2007

Christian A. Nicholes
Registration No. 50,266

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Tel: (408) 414-1224
Fax: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: MS Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

on 4/25/2007 by Annette Valdivia
Annette Valdivia

CLAIMS APPENDIX

1. A method of storing data into a database, the method comprising:
a client application receiving data;
determining one or more routines that are associated with a type of said data, wherein said one or more routines are implemented by a program that is external to both said client application and a database server that manages said database;
invoking said one or more routines;
in response to said one or more routines being invoked, said program performing steps comprising:
determining one or more first values that are specified in said data, wherein said one or more first values correspond to one or more attributes of said type; and
determining one or more second values that correspond to one or more hidden columns of one or more tables in said database;
generating, based on said one or more first values and said one or more second values, a data stream that conforms to a format of data blocks of said database; and
writing said data into one or more data blocks in said database.

2. The method of Claim 1, further comprising:
in response to said one or more routines being invoked, said program performing steps comprising:
creating a data structure that comprises:
one or more first elements that correspond to said one or more attributes; and
one or more second elements that correspond to said one or more hidden columns;
populating said one or more first elements with said one or more first values; and

populating said one or more second elements with said one or more second values;

wherein said generating of said data stream is based on said data structure.

3. The method of Claim 2, wherein said data structure is created in memory that is associated with said client application.

4. The method of Claim 1, wherein at least one of said one or more second values is associated with said one or more first values and distinguishes said one or more first values from other values in said data.

5. The method of Claim 1, wherein at least one of said one or more second values describes a position of said one or more first values relative to other values in said data.

6. The method of Claim 1, wherein a number of attributes of said type is not defined to said client application.

7. The method of Claim 1, wherein a type of an attribute of said type of said data is not defined to said client application.

8. The method of Claim 1, wherein said generating and said writing are performed without causing a Structured Query Language (SQL) engine to load said data.

9. The method of Claim 1, wherein determining said one or more routines comprises locating addresses of one or more routines that are in a same entry as an identity of said type.

10. The method of Claim 1, further comprising:

adding, to a table, an entry that indicates an association between said type and said one or more routines.

11. The method of Claim 1, further comprising:

invoking one or more routines that are located at one or more addresses that are associated with said type.

12. A method of storing data into a database, the method comprising:

a client application receiving data that conforms to a first type definition that indicates two or more first attributes, wherein at least one of said two or more first attributes is of a type that is defined by a second type definition that indicates two or more second attributes;

determining one or more first routines that are associated with said first type definition, wherein said one or more first routines are external to both said client application and a database server that manages said database;

calling said one or more first routines;

in response to one or more calls to said one or more first routines:

creating a first data structure with two or more first elements that correspond to said two or more first attributes; and

populating said two or more first elements with two or more first values that are specified in said data, wherein said two or more first values correspond to said two or more first attributes;

calling one or more second routines that are associated with said second type definition;

in response to one or more calls to said one or more second routines

creating a second data structure with two or more second elements that correspond to said two or more second attributes; and

populating said two or more second elements with two or more second values that are specified in said data, wherein said two or more second values correspond to said two or more second attributes;

generating, based on said first data structure and said second data structure, a data stream that conforms to a format of data blocks of said database; and

writing said data into one or more data blocks in said database.

13. The method of Claim 12, further comprising:

generating a set identifier that is associated with one of said one or more first elements; and

populating a plurality of elements in said second data structure with said set identifier.

14. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 1.

15. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 2.

16. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 3.

17. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 4.

18. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 5.

19. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 6.

20. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 7.

21. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 8.

22. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 9.

23. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 10.

24. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 11.

25. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 12.

26. A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 13.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

No available decisions.